

Chapter 2

Cognition

Human cognition represents a sophisticated information processing system that enables perception, reasoning, and goal-directed behavior through the orchestrated operation of multiple specialized neural circuits [98]. This cognitive architecture operates through mental states, which serve as the foundation where learning and reasoning occur. The remarkable ability to process information across different levels of abstraction and adapt to novel situations is a crucial inspiration for LLM agents [27].

The cognitive system exhibits several fundamental architectural properties reflected in Figure 1.1. First, learning functions across different mental state spaces: it can occur holistically across frontal lobes (supporting executive control and cognition) and temporal lobes (responsible for language, memory, and auditory processing), or focus on specific aspects for targeted improvement as shown by the varied research levels in the figure. Second, reasoning emerges in distinct patterns: it can follow structured templates for systematic problem-solving supported by logical reasoning and cognitive flexibility in the frontal lobes, or appear in unstructured forms for flexible thinking, particularly evident in decision-making and executive control functions. Third, the system demonstrates remarkable adaptability, continuously updating its mental states through experience while leveraging both supervised feedback (as in adaptive error correction in the cerebellum) and unsupervised environmental statistics, reflected in the different exploration stages of various cognitive functions shown in the figure [99].

These cognitive processes are supported by a modular organization, composed of distinct but interconnected components that form a cohesive system [100]. These modules include perception systems that transform raw sensory data into meaningful representations, memory systems that provide the substrate for storing and retrieving information, world models that support future scenario simulation, reward signals that guide refinement of behavior through reinforcement, emotion systems that modulate attention and resource allocation, reasoning systems that formulate decisions, and action systems that translate decisions into environmental interactions.

While human cognition implements these properties through complex neural architectures shaped by evolution, LLM agents attempt to approximate similar functions using large-scale neural models and algorithmic techniques. Understanding this biological-artificial parallel is crucial for developing more capable agents [101], as it highlights both the achievements and limitations of current systems compared to human cognition. Significant differences remain in areas such as adaptability, generalization, and contextual understanding.

In this section, we first explore **Learning**, examining both the spaces where it occurs within mental states and the specific objectives it serves. Subsequently, we investigate **Reasoning**, analyzing both structured and unstructured approaches, before concluding with a dedicated exploration of planning capabilities as a special reasoning action.

2.1 Learning

Learning represents the fundamental process through which intelligent agents transform experiences into knowledge within their mental states. This transformation occurs across different cognitive spaces, from holistic updates across the full mental state to refinement of specific cognitive components. The scope of learning encompasses remarkable capacities that serve different objectives: enhancing perceptual understanding, improving reasoning capabilities, and developing richer world understanding.

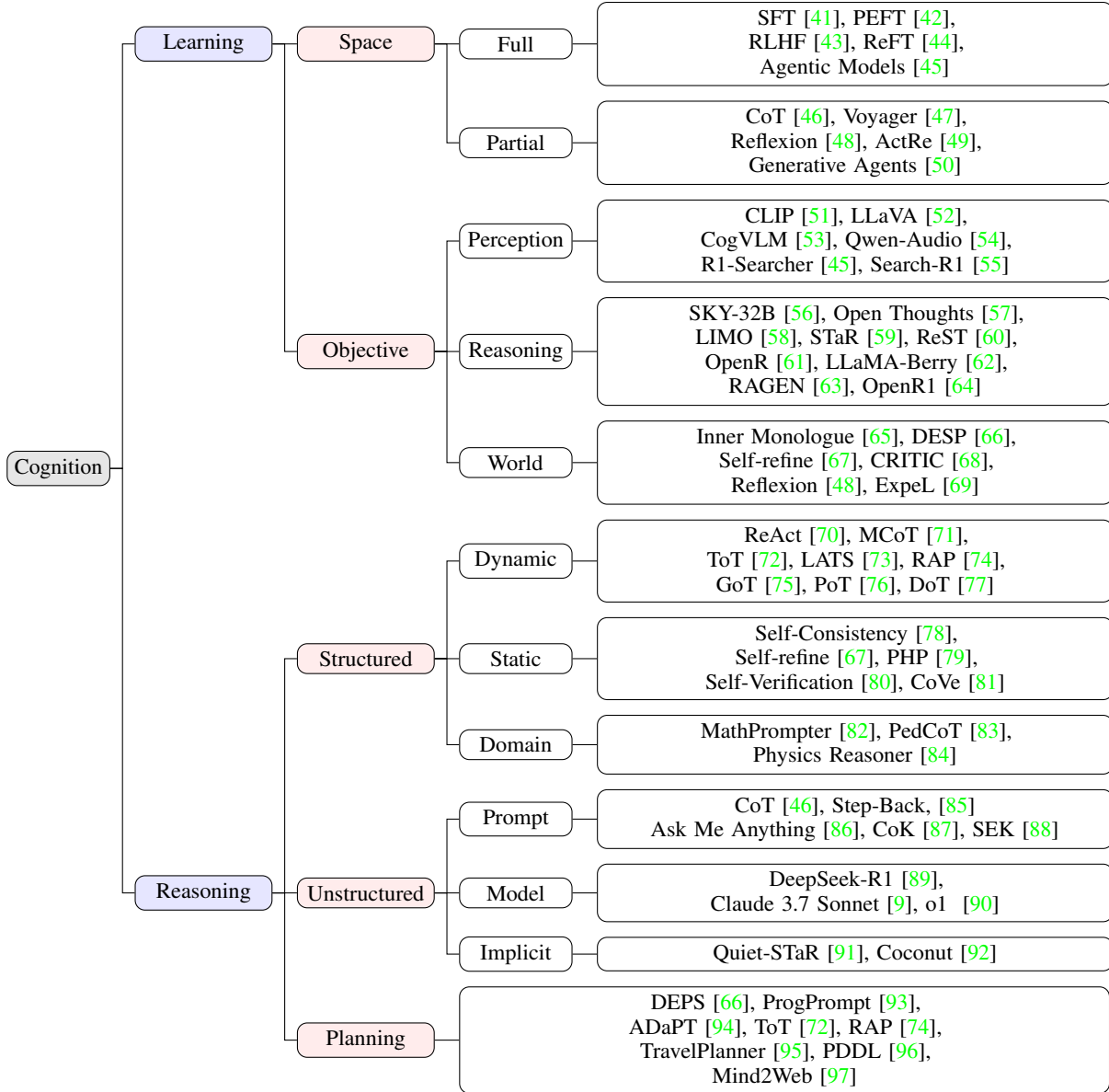


Figure 2.1: Illustrative Taxonomy of Cognition system, including learning and reasoning paradigm.

Human learning operates across multiple spaces and objectives through the brain’s adaptable neural networks. The brain coordinates learning across its entire network through integrated systems: the hippocampus facilitates rapid encoding of episodic experiences, the cerebellum supports supervised learning for precise motor skills, the basal ganglia enable reinforcement learning through dopaminergic reward signals, and cortical regions facilitate unsupervised pattern extraction [99]. At more focused levels, specific neural circuits can undergo targeted adaptation, allowing for specialized skill development and knowledge acquisition. These systems work together on different timescales, ranging from immediate responses to lifelong development, while being influenced by factors like attention, emotions, and social environment [27].

LLM agents, while fundamentally different in architecture, implement analogous learning processes across their mental state spaces. At the comprehensive level, they acquire broad knowledge through pre-training on massive datasets, demonstrating a form of unsupervised learning. At more focused levels, they refine specific capabilities through parameter-updating mechanisms like supervised fine-tuning and reinforcement learning. Uniquely, they also demonstrate in-context learning capabilities, adapting to novel tasks without parameter changes by leveraging context

within their attention window: a capability that mirrors aspects of human working memory but operates through fundamentally different mechanisms.

The comparison between human and artificial learning systems provides valuable insights for developing more capable, adaptive agents. Human learning demonstrates notable characteristics in efficiency, contextualization, and integration with emotional systems, while LLM-based approaches show distinct capabilities in processing large datasets, representing formal knowledge, and synthesizing information across domains. These complementary strengths suggest productive directions for research. As we explore the foundations of learning, we first examine the spaces where learning occurs within mental states, followed by an analysis of the specific objectives that drive learning processes.

Table 2.1: Summary of Learning Methods with Different State Modifications. ● indicates primary impact while ○ indicates secondary or no direct impact.

Method	Model	Perception	Reasoning	Memory	Reward	World Model
Voyager [47]	○	○	○	●	○	○
Generative Agents [50]	○	○	○	●	○	○
Learn-by-interact [102]	●	○	○	●	○	○
RAGEN [63]	●	○	●	○	●	○
DigiRL [103]	●	○	●	○	●	○
R1-Searcher [45]	●	●	●	○	●	○
RewardAgent [104]	●	○	○	○	●	○
Text2Reward [105]	○	○	○	○	●	○
ARAMP [106]	●	○	○	○	●	○
ActRe [49]	●	○	●	○	○	●
WebDreamer [107]	○	○	○	○	○	●
RAP [74]	○	○	○	○	○	●
AutoManual [108]	○	○	○	●	○	●

2.1.1 Learning Space

The learning approaches in LLM agents represent a structured, data-driven paradigm in contrast to the exploratory, emotionally-driven learning observed in humans. While human learning often involves active curiosity, motivation, and emotional reinforcement, LLM-based agents typically learn through more formalized processes, such as parameter updates during training or structured memory formation during exploration. Current agent architectures attempt to bridge this gap by implementing mechanisms that simulate aspects of human learning while leveraging the strengths of computational systems.

Learning within an intelligent agent occurs across different spaces, encompassing both the underlying model θ and mental states M , where the former fundamentally supports the capabilities and limitations of the latter. Formally, we define an intelligent agent’s internal state as a tuple $\mathcal{I} = (\theta, M)$ that includes both the model parameters and mental state components. The mental state can be further decomposed into different structures as we illustrated in 1.2:

$$M = \{M^{mem}, M^{wm}, M^{emo}, M^{goal}, M^{rew}\} \quad (2.1)$$

where M^{mem} represents memory, M^{wm} denotes world model, M^{emo} indicates emotional state, M^{goal} represents goals, and M^{rew} represents reward signals.

Modifications to the underlying model can be viewed as full mental state learning, as they fundamentally alter the agent’s capabilities. While model-level modifications can affect different mental states to varying degrees, changes to the model’s context window or external structures tend to focus on specific mental state components. For instance, learning experiences and skills from the environment primarily influence memory, while leveraging the LLM’s inherent predictive capabilities enhances the world model.

Full Mental State Learning Full mental state learning enhances the capabilities of an agent through comprehensive modifications to the underlying model θ , which in turn affects all components of the mental state M . This process begins with pre-training, which establishes the foundation of language models by acquiring vast world knowledge, analogous to how human babies absorb environmental information during development, though in a more structured and extensive manner.

Post-training techniques represent the cornerstone for advancing agent capabilities. Similar to how human brains are shaped by education, these techniques while affecting the entire model, can emphasize different aspects of cognitive

development. Specifically, various forms of tuning-based learning enable agents to acquire domain-specific knowledge and logical reasoning capabilities. Supervised Fine-Tuning (SFT) [41] serves as the fundamental approach where models learn from human-labeled examples, encoding knowledge directly into the model’s weights. For computational efficiency, Parameter-Efficient Fine-Tuning (PEFT) methods have emerged. Adapter-BERT [42] introduced modular designs that adapt models to downstream tasks without modifying all parameters, while Low-Rank Adaptation (LoRA) [109] achieves similar results by decomposing weight updates into low-rank matrices, adjusting only a small subset of effective parameters.

Some agent capabilities are closely connected to how well they align with human preferences, with alignment-based learning approaches modifying the model to reshape aspects of the agent’s underlying representations. Reinforcement learning from human feedback (RLHF) [110] aligns models with human values by training a reward model on comparative judgments and using this to guide policy optimization. InstructGPT [43] demonstrated how this approach could dramatically improve consistency with user intent across diverse tasks. Direct Preference Optimization (DPO) [111] has further simplified this process by reformulating it as direct preference learning without explicit reward modeling, maintaining alignment quality while reducing computational complexity.

Reinforcement learning (RL) presents a promising pathway for specialized learning in specific environments. RL has shown particular promise in enhancing reasoning capabilities, essentially enabling the agent’s underlying model to learn within the space of thought. Foundational works such as Reinforcement Fine-Tuning (ReFT) [44] enhance reasoning through fine-tuning with automatically sampled reasoning paths under online reinforcement learning rewards. DeepSeek-R1 [89] advances this approach through rule-based rewards and Group Relative Policy Optimization (GRPO) [112], while Kimi k1.5 [113] combines contextual reinforcement learning with optimized chain-of-thought techniques to improve both planning processes and inference efficiency. In specific environments, modifying models to enhance agents’ understanding of actions and external environments has proven effective, as demonstrated by DigiRL [103], which implements a two-stage reinforcement learning approach enabling agents to perform diverse commands on real-world Android device simulators.

Recent works have attempted to integrate agent action spaces directly into model training [45, 55], enabling learning of appropriate actions for different states through RL or SFT methods. This integration fundamentally affects the agent’s memory, reward understanding, and world model comprehension, pointing toward a promising direction for the emergence of agentic models.

Partial Mental State Learning While full mental state learning through model modifications provides comprehensive capability updates, learning focused on particular components of an agent’s mental state M represents another essential and often more efficient approach. Such partial mental state learning can be achieved either through targeted model updates or through in-context adaptation without parameter changes.

In-Context Learning (ICL) illustrates how agents can effectively modify specific mental state components without modifying the entire model. This mechanism allows agents to adapt to new tasks by leveraging examples or instructions within their context window, paralleling human working memory’s role in rapid task adaptation. Chain-of-Thought (CoT) [46] demonstrates the effectiveness of this approach, showing how agents can enhance specific cognitive capabilities while maintaining their base model parameters unchanged.

The feasibility of partial mental state learning is evidenced through various approaches targeting different components such as memory (M^{mem}), reward (M^{rew}), and world model (M^{wm}). Through normal communication and social interaction, Generative Agents [50] demonstrate how agents can accumulate and replay memories, extracting high-level insights to guide dynamic behavior planning. In environmental interaction scenarios, Voyager [47] showcases how agents can continuously update their skill library through direct engagement with the Minecraft environment, accumulating procedural knowledge without model retraining. Learn-by-Interact [102] further extends this approach by synthesizing experiential data through direct environmental interaction, eliminating the need for manual annotation or reinforcement learning frameworks. Additionally, agents can learn from their mistakes and improve through reflection, as demonstrated by Reflexion [48], which guides agents’ future thinking and actions by obtaining textual feedback from repeated trial and error experiences.

Modifications to reward and world models provide another example of partial mental state learning. ARMAP [106] refines environmental reward models by distilling them from agent action trajectories, providing a foundation for further learning. AutoMC [114] constructs dense reward models through environmental exploration to support agent behavior. Meanwhile, [107] explicitly leverages LLMs as world models to predict the impact of future actions, effectively modifying the agent’s world understanding (M^{wm}). ActRe[49] builds upon the language model’s inherent world understanding to construct tasks from trajectories, enhancing the agent’s capabilities as both a world model and reasoning engine through iterative training.

2.1.2 Learning Objective

The learning process of intelligent agents manifests across all aspects of their interaction with the environment. At the input level, agents learn to better perceive and parse environmental information; at the processing level, agents learn how to conduct effective reasoning based on existing knowledge or reasoning capabilities; at the comprehension level, agents form and optimize their understanding of the world through continuous interaction. This multi-level learning objective framework enables agents to evolve continuously across different dimensions, allowing them to better handle complex and dynamic task environments.

Learning for Better Perception The ability to effectively perceive and process information from the environment is fundamental to agent intelligence. To enhance perceptual capabilities, agents employ two primary learning approaches: expanding multimodal perception and leveraging retrieval mechanisms.

Multimodal perception learning enables agents to process and integrate diverse sensory inputs, similar to human multi-sensory integration but unconstrained by biological limitations. This capability has evolved significantly through advances like CLIP [51], which pioneered the alignment of visual and linguistic representations in shared embedding spaces. Building on this foundation, models like LLaVA [52] enhanced visual perception by training specialized projectors on image-text pairs, while CogVLM [53] advanced visual reasoning through unified representational architectures.

The expansion of perceptual modalities continues across multiple sensory domains. In audio processing, Qwen-Audio [54] demonstrates the unified encoding of diverse acoustic information, from speech to environmental sounds. Recent work by [115] has even ventured into tactile perception, developing datasets that align touch, vision, and language representations. These advances enable agents to engage more comprehensively with both physical and digital environments.

Agents also learn to enhance their observational capabilities through retrieval mechanisms. Unlike human perception, which is constrained by immediate sensory input, agents can learn to access and integrate information from vast external knowledge repositories. Retrieval-augmented approaches like RAG [116] enhance perceptual understanding by connecting immediate observations with relevant stored knowledge.

Recent work on retrieval-based agents demonstrates the potential for enhancing active information acquisition capabilities. Search-o1 [117] guides reasoning models to learn active retrieval through prompting, thereby expanding their knowledge boundaries. Taking this further, R1-Searcher [45] and Search-R1 [55] directly incorporate retrieval capabilities into the model, enabling autonomous information retrieval during the reasoning process. These advances suggest a promising direction for improving agent perception: enhancing model-level active perception capabilities to enrich the foundation for decision-making. This approach may represent a significant avenue for future agent development.

Learning for Better Reasoning Reasoning serves as a critical bridge between an agent’s mental state and its actions, making the ability to reason effectively and the development of reasoning capabilities essential for intelligent agents. The foundation of reasoning in modern agents stems from two key elements: the rich world knowledge embedded in their underlying models, and the robust logical frameworks supported either internally or through context structuring. This makes learning for better reasoning a vital objective in agent development.

The development of reasoning capabilities is demonstrated through several key phenomena. First, high-quality reasoning data directly enhances model reasoning ability; second, such high-quality data often requires verification or reward models for effective curation; and third, direct reinforcement learning on foundation models can spontaneously manifest reasoning capabilities.

The importance of reasoning in agent development has been re-emphasized following the release of the o1 series. A common approach involves collecting and distilling data from open/closed-source reasoning models. For instance, SKY-32B [56] distilled data from QWQ-32B [118] to train a 32B reasoning model at a cost of \$450. Similarly, Open Thoughts [57] trained Bespoke-Stratos-32B at a low cost by distilling and synthesizing datasets from R1. These studies demonstrate that even without complex algorithmic design, using reasoning data to perform Supervised Fine-Tuning (SFT) on base models can effectively activate reasoning capabilities.

Another crucial insight regarding data quality is that highly structured reasoning data more effectively enables agents and language models to learn reasoning processes. Notably, LIMO [58] demonstrated that powerful reasoning models could be built with extremely few data samples by constructing long and effective reasoning chains for complex reasoning tasks. This insight stems from their observation that language models inherently possess sufficient knowledge for reasoning but require high-quality reasoning paths to activate these capabilities. Supporting this view, Li et al.

[119] revealed that both Long CoT and Short CoT fundamentally teach models to learn reasoning structures rather than specific content, suggesting that automated selection of high-quality reasoning data may become an important future direction.

One viable exploration approach involves first conducting extensive searches, and then using verifiable environments or trainable reward models to provide feedback on reasoning trajectories, thereby filtering out high-quality reasoning data. This approach has led to several families of techniques that leverage different feedback mechanisms to improve reasoning capabilities.

The first category follows the bootstrap paradigm exemplified by STaR [59] and its variants, which implement techniques where models generate step-by-step rationales and iteratively improve through fine-tuning on successful reasoning paths. This family includes Quiet-STaR [91], V-STaR [120], and rStar-Math [121], with the latter specifically enhancing mathematical reasoning through reinforcement learning principles. By iteratively selecting correct reasoning paths for training, these methods achieve self-improvement through successive refinement cycles.

The second category extends this paradigm by more explicitly incorporating reinforcement learning principles. The ReST family, beginning with the original ReST [60] introducing reinforced self-training, performs multiple attempts (typically 10) per sample and creates new training datasets from successful reasoning instances. ReST-EM [122] enhances the approach with expectation maximization, while ReST-MCTS [122] further integrates Monte Carlo Tree Search to enable improved reasoning capabilities through more sophisticated exploration strategies.

Several approaches have introduced Policy Reward Models (PRMs) to provide quality feedback on reasoning paths. Methods like OpenR [61] and LLaMA-Berry [62] model reasoning tasks as Markov Decision Processes (MDPs) and leverage tree search to explore diverse reasoning paths while using PRMs for quality assessment. In domain-specific applications, methods like rStar-Math [121] and DeepSeekMath [112] have demonstrated success in mathematical problem-solving through multi-round self-iteration and balanced exploration-exploitation strategies. For code generation, o1-Coder [123] leverages MCTS to generate code with reasoning processes, while Marco-o1 [123] extends this approach to open-ended tasks. These implementations highlight how the synergy between MCTS and PRM achieves effective reasoning path exploration while maintaining solution quality through fine-grained supervision.

Beyond data-driven approaches, reinforcement learning (RL) has demonstrated remarkable success in enhancing language models' reasoning capabilities, as evidenced by recent breakthroughs like DeepSeek R1 [89] and Kimi-K1.5 [113]. The foundation of RL for LLMs can be traced to several pioneering frameworks: ReFT [44] introduced a combination of supervised fine-tuning and online reinforcement learning, while VeRL [124] established an open-source framework supporting various RL algorithms for large-scale models up to 70B parameters. RFT [125] further demonstrated the effectiveness of reward-guided optimization in specific reasoning tasks.

Building upon these foundations, subsequent works have explored diverse applications and improvements. OpenR1 [64] and RAGEN [63] extended RL techniques to enhance general reasoning capabilities, while specialized implementations like SWE-Gym [126] demonstrated success in software engineering tasks. Notably, DigiRL [103] introduced novel approaches for digital-world agent enhancement.

Recent advances have further integrated RL with tool usage and reasoning. Qwen-QwQ-32B [118] employs reinforcement learning and a general reward mechanism to incorporate tool calling into the reasoning process, enabling the seamless use of arbitrary tools during reasoning and achieving agent-like capabilities directly within the model. Similarly, RAGEN [63] focuses on multi-step agentic scenarios, establishing a framework for agent reinforcement learning in complex environments. These developments suggest an increasing convergence between model training and agent development, potentially leading to more integrated and capable intelligent systems. These implementations highlight how RL can effectively improve model performance while reducing dependence on large-scale annotated datasets, particularly in complex reasoning scenarios.

Learning for World Understanding A critical aspect of agent intelligence is the ability to understand how the world operates through direct interaction and experience accumulation. This understanding encompasses how the environment responds to different actions and the consequences these actions bring. Through continuous interaction with their environment, agents can build and refine their *memory*, *reward understanding*, and *world model*, learning from both successes and failures to develop a more comprehensive grasp of their operational domain.

Recent research has revealed diverse approaches to experiential learning for world understanding. At the foundational level, Inner Monologue [65] demonstrates how agents can accumulate basic environmental knowledge through continuous interaction. Similarly, Learn-by-Interact [102] shows that meaningful understanding can emerge from direct environmental engagement without explicit reward mechanisms. More sophisticated approaches are exemplified by DESP [66] and Voyager [47] in the Minecraft environment, where agents not only gather experiences but also actively process them: DESP through outcome analysis and Voyager through dynamic skill library expansion.

The processing and utilization of accumulated experiences have been further systematized through advanced frameworks. Generative Agents [50] introduces sophisticated memory replay mechanisms, enabling agents to extract high-level insights from past interactions. This systematic approach is enhanced by Self-refine [67] and Critic [68], which implement structured cycles of experience evaluation and refinement.

The optimization of reward understanding through environmental interaction has emerged as another crucial aspect of world understanding. Text2Reward [105] demonstrates how agents can continuously refine reward functions through human feedback, better aligning them with task objectives and environmental characteristics. Similarly, AutoManual [108] builds behavioral guidelines through sustained interaction, developing reward-verified protocols that provide a foundation for understanding environmental rewards and decision-making. These interaction-based optimization mechanisms enable agents to better comprehend environmental dynamics and generate more precise reward signals, ultimately enhancing their adaptability and decision-making capabilities in complex, dynamic environments.

Building on these foundations, RAP [74] represents a significant advancement by conceptualizing reasoning as planning with a world model. By repurposing LLMs as both reasoning agents and world models, RAP enables agents to simulate the outcomes of potential actions before committing to them, facilitating more effective planning through Monte Carlo Tree Search. This approach allows agents to strategically explore the reasoning space with a proper balance between exploration and exploitation.

Further innovations in leveraging world models for agent learning include ActRe [127], which reverses the typical reasoning-action sequence by first performing actions and then generating post-hoc explanations. This capability to rationalize actions demonstrates LLMs’ inherent understanding of world dynamics, enabling autonomous trajectory annotation and facilitating contrastive self-training.

The importance of cognitive maps in world understanding is highlighted by [128], who show that structured mental representations inspired by human cognition significantly enhance LLMs’ extrapolation capabilities in novel environments. These cognitive maps not only improve planning but also exhibit human-like characteristics such as structured mental simulation and rapid adaptation.

In web-based environments, recent work by [107] and [129] demonstrates that LLMs can function as effective world models for anticipating the outcomes of web interactions. By simulating potential state changes before executing actions, these approaches enable safer and more efficient decision-making, particularly in environments where actions may be irreversible.

Through systems like Reflexion [48] and ExpeL [69], agents have advanced experiential learning by autonomously managing the full cycle of experience collection, analysis, and application, enabling them to learn effectively from both successes and failures.

These developments collectively illustrate how world models are becoming increasingly central to agent learning systems, providing a foundation for understanding environmental dynamics and enabling more effective planning, reasoning, and decision-making in complex, interactive environments.

2.2 Reasoning

Reasoning represents the key to intelligent behavior, transforming raw information into actionable knowledge that drives problem-solving and decision-making. For both humans and artificial agents, it enables logical inference, hypothesis generation, and purposeful interaction with the world. In human cognition, reasoning emerges through multiple strategies: deductive reasoning applies general rules to specific cases, inductive reasoning builds generalizations from particular instances, and abductive reasoning constructs plausible explanations from incomplete data [130, 131]. These processes are augmented by heuristics—mental shortcuts that streamline decision-making under uncertainty—and are continuously refined through environmental feedback, ensuring that reasoning remains grounded in reality and adaptive to change.

For LLM-based agents, reasoning serves a parallel role, elevating them beyond reactive systems to proactive entities capable of sophisticated cognition. Through reasoning, these agents process multimodal inputs, integrate diverse knowledge sources, and formulate coherent strategies to achieve objectives. The environment plays a dual function: supplying information that fuels reasoning and serving as the proving ground where reasoned actions are tested, creating a feedback loop that enables agents to validate inferences and learn from errors.

In LLM-based agents, reasoning can be formally defined as the process of action selection based on mental states, representing a crucial bridge between perception and action. More precisely, given a mental state M_t at time t , reasoning can be formalized as a function $R(M_t) \rightarrow a_t$, where a_t represents the selected action. This process operates across

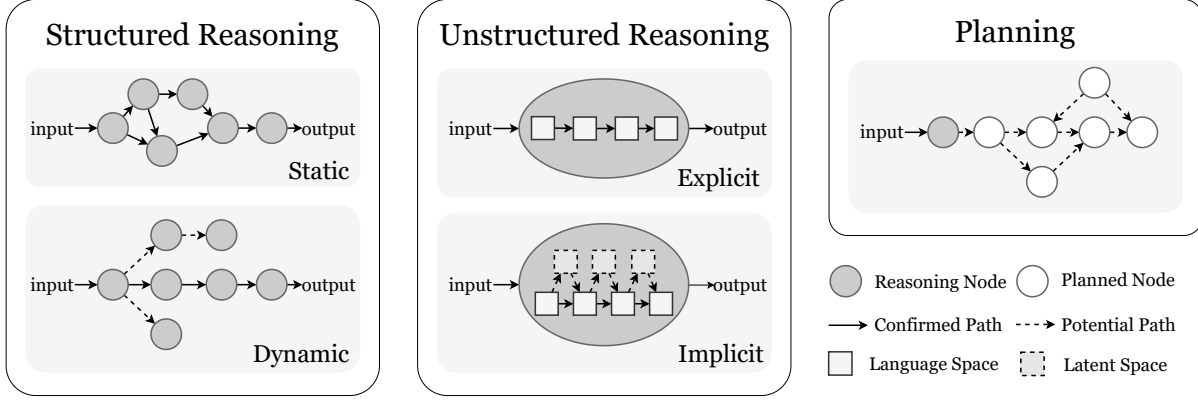


Figure 2.2: Comparison of reasoning paradigms in LLM-based agents.

various environments—textual, digital, and physical worlds—where completing a task typically requires either a single reasoning step or a composition of multiple reasoning actions.

The composition of reasoning actions naturally leads to two distinct approaches: structured and unstructured reasoning. Structured reasoning (R_s) can be formalized as an explicit composition $R_s = R_1 \circ R_2 \circ \dots \circ R_n$, where each R_i represents a discrete reasoning step with clear logical dependencies. In contrast, unstructured reasoning (R_u) takes a more holistic form $R_u = f(M_t)$, where the composition remains implicit and flexible, allowing for dynamic adaptation to context. This dual framework mirrors human cognition, where structured reasoning parallels our explicit logical deduction processes, while unstructured reasoning reflects our capacity for intuitive problem-solving and pattern recognition.

The environment plays a crucial role in this formalization, serving both as a source of observations o_t that influence mental state updates ($M_t = L(M_{t-1}, a_{t-1}, o_t)$) and as a testing ground for reasoning outcomes. This creates a continuous feedback loop where reasoning not only drives action selection but also influences how the agent’s mental state evolves, enabling iterative refinement of reasoning strategies through experience.

In this section, we will examine how these reasoning approaches manifest in practice. We begin with structured reasoning, which emphasizes systematic problem decomposition and multi-step logical chains. We then explore unstructured reasoning, which allows for flexible response patterns and parallel solution exploration. Finally, we investigate planning as a specialized form of reasoning that combines both structured and unstructured approaches for tackling complex, long-horizon tasks.

2.2.1 Structured Reasoning

Structured reasoning represents a methodical approach to problem-solving that employs explicit organizational frameworks to guide the reasoning process. Unlike unstructured approaches, structured reasoning makes the composition of reasoning steps explicit, which can be formalized as $R_s = R_1 \circ R_2 \circ \dots \circ R_n$, where each R_i represents a discrete reasoning step with clear logical dependencies. In this formulation, each reasoning node is an explicitly executed computational unit, and the connections between nodes represent definite information flow paths. This approach enables more systematic exploration of solution spaces and facilitates more robust decision-making through deliberate step-by-step analysis, providing high interpretability and traceability throughout the reasoning process.

2.2.1.1 Dynamic Reasoning Structures

Dynamic reasoning structures allow for the adaptive construction of reasoning paths during problem-solving, creating versatile frameworks that can adjust based on intermediate results and insights.

Linear Sequential Reasoning Linear structures frame reasoning as a series of sequential steps, where each step builds on the one before. ReAct [70] illustrates this by combining reasoning traces with task-specific actions in an alternating fashion. This combination allows for reasoning traces to guide and modify action plans while actions can access external sources for further information. This mutual interaction improves both reasoning integrity and environmental adaptation.

Reasoning via Planning (RAP) [74] extends the linear reasoning paradigm by formulating LLM reasoning as a Markov decision process, though it was limited by states specifically designed for particular problems. The Markov Chain of Thought (MCoT) [71] extended this paradigm by conceptualizing each reasoning step as a Markovian state accompanied by executable code. This approach enables efficient next-step inference without requiring a lengthy context window by compressing previous reasoning into a simplified math question. Atom of Thoughts [132] explicitly defined problems as state representations and designed a general decomposition-contraction two-phase state transition mechanism to construct Markovian reasoning processes, transforming complex problems into a series of atomic questions.

Tree-Based Exploration Tree-based approaches expand beyond linear structures by organizing reasoning into hierarchical frameworks that support branching exploration. Tree of Thoughts (ToT) [72] introduces a structured approach where complex problems are decomposed into intermediate steps, enabling breadth-first or depth-first search through the solution space. This allows the model to consider multiple reasoning paths simultaneously and systematically explore alternatives.

Language Agent Tree Search (LATS) [73] advances this paradigm by integrating Monte Carlo Tree Search (MCTS) with LLMs, using the environment as an external feedback mechanism. This approach enables more deliberate and adaptive problem-solving by balancing exploration and exploitation through a sophisticated search process guided by LLM-powered value functions and self-reflection.

Reasoning via Planning (RAP) [74] further enhances tree-based reasoning by repurposing LLMs as both reasoning agents and world models. Through this dual role, RAP enables agents to simulate the outcomes of potential reasoning paths before committing to them, creating a principled planning framework that balances exploration with exploitation in the reasoning space.

Graph-Based Reasoning Graph structures offer even greater flexibility by allowing non-hierarchical relationships between reasoning steps. Graph of Thoughts (GoT) [75] extends tree-based approaches to arbitrary graph structures, enabling more complex reasoning patterns that can capture interdependencies between different steps. This approach allows for connections between seemingly disparate reasoning branches, facilitating more nuanced exploration of the solution space.

Path of Thoughts (PoT) [76] addresses relation reasoning challenges by decomposing problems into three key stages: graph extraction, path identification, and reasoning. By explicitly extracting a task-agnostic graph that identifies entities, relations, and attributes within the problem context, PoT creates a structured representation that facilitates the identification of relevant reasoning chains, significantly improving performance on tasks requiring long reasoning chains.

Diagram of Thought (DoT) [77] models iterative reasoning as the construction of a directed acyclic graph (DAG), organizing propositions, critiques, refinements, and verifications into a unified structure. This approach preserves logical consistency while enabling the exploration of complex reasoning pathways, providing a theoretically sound framework grounded in Topos Theory.

2.2.1.2 Static Reasoning Structures

Static reasoning structures employ fixed frameworks that guide the reasoning process without dynamically adjusting the structure itself, focusing instead on improving the content within the established framework.

Ensemble Methods. Ensemble approaches leverage multiple independent reasoning attempts to improve overall performance through aggregation. Self-Consistency [78] pioneered this approach by sampling multiple reasoning paths rather than relying on single greedy decoding, significantly improving performance through majority voting among the generated solutions.

MedPrompt [133] demonstrates how domain-specific ensemble techniques can enhance performance by carefully crafting prompts that elicit diverse reasoning approaches, achieving state-of-the-art results on medical benchmarks through systematic composition of prompting strategies.

LLM-Blender [134] introduces a sophisticated ensembling framework that leverages the diverse strengths of multiple LLMs through pairwise comparison (PairRanker) and fusion (GenFuser) of candidate outputs. This approach enables the system to select the optimal model output for each specific example, creating responses that exceed the capabilities of any individual model.

Progressive Improvement. Progressive improvement frameworks focus on iteratively refining reasoning through structured feedback loops. Self-Refine [67] implements an iterative approach where the model generates initial output, provides self-feedback, and uses that feedback to refine itself. This mimics human revision processes without requiring additional training or reinforcement learning, resulting in significant improvements across diverse tasks.

Reflexion [48] extends this concept by integrating environmental feedback, enabling agents to verbally reflect on task feedback signals and maintain reflective text in an episodic memory buffer. This approach guides future decision-making by incorporating insights from previous attempts, significantly enhancing performance in sequential decision-making, coding, and reasoning tasks.

Progressive-Hint Prompting (PHP) [79] further develops this paradigm by using previously generated answers as hints to progressively guide the model toward correct solutions. This approach enables automatic multiple interactions between users and LLMs, resulting in significant accuracy improvements while maintaining high efficiency.

Error Correction. Error correction frameworks focus specifically on identifying and addressing mistakes in the reasoning process. Self-Verification [80] introduces a self-critique system that enables models to backward-verify their conclusions by taking the derived answer as a condition for solving the original problem, producing interpretable validation scores that guide answer selection.

Refiner [135] addresses the challenge of scattered key information by adaptively extracting query-relevant content and restructuring it based on interconnectedness, highlighting information distinction and effectively aligning downstream LLMs with the original context.

Chain-of-Verification (CoVe) [81] tackles factual hallucinations through a structured process where the model drafts an initial response, plans verification questions, independently answers those questions, and generates a final verified response. This deliberate verification process significantly reduces hallucinations across a variety of tasks.

Recursive Criticism and Improvement (RCI) [128] enables LLMs to execute computer tasks by recursively criticizing and improving their outputs, outperforming existing methods on the MiniWoB++ benchmark with only a handful of demonstrations per task and without task-specific reward functions.

Critic [68] extends this approach by integrating external tools for validation, enabling LLMs to evaluate and progressively amend their outputs like human interaction with tools. This framework allows initially “black box” models to engage in a continuous cycle of evaluation and refinement, consistently enhancing performance across diverse tasks.

2.2.1.3 Domain-Specific Reasoning Frameworks

Domain-specific reasoning frameworks adapt structured reasoning approaches to the unique requirements of particular domains, leveraging specialized knowledge and techniques to enhance performance in specific contexts.

MathPrompter [82] addresses arithmetic reasoning challenges by generating multiple algebraic expressions or Python functions to solve the same math problem in different ways. This approach improves confidence in the output results by providing multiple verification paths, significantly outperforming state-of-the-art methods on arithmetic benchmarks.

Physics Reasoner [84] addresses the unique challenges of physics problems through a knowledge-augmented framework that constructs a comprehensive formula set and employs detailed checklists to guide effective knowledge application. This three-stage approach—problem analysis, formula retrieval, and guided reasoning—significantly improves performance on physics benchmarks by mitigating issues of insufficient knowledge and incorrect application.

Pedagogical Chain-of-Thought (PedCoT) [83] leverages educational theory, particularly the Bloom Cognitive Model, to guide the identification of reasoning mistakes in mathematical contexts. This approach combines pedagogical principles for prompt design with a two-stage interaction process, providing a foundation for reliable mathematical mistake identification and automatic answer grading.

The evolution of structured reasoning in LLM agents reflects a growing understanding of how to enhance reasoning capabilities through explicit organizational frameworks. From linear sequences to complex graphs, and ensemble methods to specialized domain frameworks, these approaches demonstrate the power of structural guidance in improving reasoning performance across diverse tasks and domains.

2.2.2 Unstructured Reasoning

In contrast to structured reasoning approaches that explicitly organize reasoning steps, unstructured reasoning (R_u) takes a holistic form $R_u = f(M_t)$, where the composition remains implicit and flexible. In this mode, the reasoning process is encapsulated within a single function mapping, without explicitly defining intermediate steps or state transitions. This approach leverages the inherent capabilities of language models to generate coherent reasoning without enforcing rigid structural constraints, with intermediate reasoning processes occurring explicitly in the language space or implicitly in the latent space. Unstructured reasoning methods have demonstrated remarkable effectiveness across diverse tasks while maintaining simplicity and efficiency in implementation.

2.2.2.1 Prompting-Based Reasoning

The most accessible way to elicit reasoning in LLM agents lies in carefully crafted prompts. By providing appropriate reasoning demonstrations or instructing LLMs to perform inferential steps, agents can leverage their logical deduction capabilities to solve problems through flexible reasoning processes.

Chain-of-Thought Variants. The cornerstone of prompting-based reasoning is Chain-of-Thought (CoT) prompting [46], which operationalizes reasoning through few-shot examples with explicit generation of intermediate rationalization steps. This foundational technique has inspired several evolutionary variants that enhance its basic approach. Zero-shot CoT [136] eliminates the need for demonstration examples through strategic prompting (e.g., “Let’s think step by step”), making the approach more accessible while maintaining effectiveness. Auto-CoT [137] automates the creation of effective demonstrations by clustering diverse questions and generating reasoning chains for representative examples from each cluster. Least-to-Most Prompting [138] addresses complex reasoning by decomposing problems into sequential sub-problems, enabling a progressive planning process that facilitates easy-to-hard generalization. Complex CoT [139] further enhances reasoning depth by specifically selecting high-complexity exemplars as prompting templates, better-equipping models to tackle intricate problems.

Problem Reformulation Strategies. Advanced prompting strategies demonstrate architectural innovations in reasoning guidance by reformulating the original problem. Step-Back Prompting [85] implements abstraction-first reasoning through conceptual elevation, enabling models to derive high-level concepts and first principles before addressing specific details. Experimental results demonstrate substantial performance gains on various reasoning-intensive tasks, with improvements of 7-27% across physics, chemistry, and multi-hop reasoning benchmarks. Rephrase and Respond [140] employ semantic expansion to transform original questions into more tractable forms, allowing models to approach problems from multiple linguistic angles and identify the most effective problem formulation.

Abstraction-of-Thought [141] introduces a novel structured reasoning format that explicitly requires varying levels of abstraction within the reasoning process. This approach elicits language models to first contemplate at the abstract level before incorporating concrete details, a consideration overlooked by step-by-step CoT methods. By aligning models with the AoT format through finetuning on high-quality samples, the approach demonstrates substantial performance improvements across a wide range of reasoning tasks compared to CoT-aligned models.

Enhanced Prompting Frameworks. Several frameworks extend the basic prompting paradigm to create more sophisticated reasoning environments. Ask Me Anything [86] constrains open-ended generation by reformulating tasks into structured question-answer sequences, enforcing focused reasoning trajectories. This approach recursively uses the LLM itself to transform task inputs to the effective QA format, enabling open-source GPT-J-6B to match or exceed the performance of few-shot GPT3-175B on 15 of 20 popular benchmarks.

Algorithm of Thoughts [142] proposes a novel strategy that propels LLMs through algorithmic reasoning pathways by employing algorithmic examples fully in context. This approach exploits the innate recurrence dynamics of LLMs, expanding their idea exploration with merely one or a few queries. The technique outperforms earlier single-query methods and even more recent multi-query strategies while using significantly fewer tokens, suggesting that instructing an LLM using an algorithm can lead to performance surpassing that of the algorithm itself.

Chain-of-Knowledge (CoK) [87] augments LLMs by dynamically incorporating grounded information from heterogeneous sources, resulting in more factual rationales and reduced hallucination. CoK consists of three stages: reasoning preparation, dynamic knowledge adapting, and answer consolidation, leveraging both unstructured and structured knowledge sources through an adaptive query generator. This approach corrects rationales progressively using preceding corrected rationales, minimizing error propagation between reasoning steps.

Self-Explained Keywords (SEK) [88] addresses the challenge of low-frequency terms in code generation by extracting and explaining key terms in problem descriptions with the LLM itself and ranking them based on frequency. This approach significantly improves code generation performance across multiple benchmarks, enabling models to shift attention from low-frequency keywords to their corresponding high-frequency counterparts.

2.2.2.2 Reasoning Models

Recent advances in language models have led to the development of specialized reasoning models designed explicitly for complex inferential tasks. These models are fine-tuned or specially trained to optimize reasoning capabilities, incorporating architectural and training innovations that enhance their performance on tasks requiring multi-step logical inference.

Reasoning models like DeepSeek’s R1 [89], Anthropic’s Claude 3.7 Sonnet [9], and OpenAI’s o series models [90] represent the frontier of reasoning capabilities, demonstrating remarkable proficiency across diverse reasoning bench-

marks. These models are trained with specialized methodologies that emphasize reasoning patterns, often incorporating significant amounts of human feedback and reinforcement learning to enhance their inferential abilities.

The emergence of dedicated reasoning models reflects a growing understanding of the importance of reasoning capabilities in language models and the potential benefits of specialized training for these tasks. By concentrating on reasoning-focused training data and objectives, these models achieve performance levels that significantly exceed those of general-purpose language models, particularly on tasks that require complex logical inference, mathematical reasoning, and multi-step problem-solving.

2.2.2.3 Implicit Reasoning

Beyond explicit reasoning approaches, recent research has explored the potential of implicit reasoning methods that operate without overtly exposing the reasoning process. These approaches aim to improve efficiency by reducing the number of tokens generated while maintaining or enhancing reasoning performance.

Quiet-STaR [91] generalizes the Self-Taught Reasoner approach by teaching LMs to generate rationales at each token to explain the future text, improving their predictions. This approach addresses key challenges including computational cost, the initial unfamiliarity with generating internal thoughts, and the need to predict beyond individual tokens. Experimental results demonstrate zero-shot improvements in mathematical reasoning (5.9%→10.9%) and commonsense reasoning (36.3%→47.2%) after continued pretraining, marking a step toward LMs that learn to reason in a more general and scalable way.

Chain of Continuous Thought (Coconut) [92] introduces a paradigm that enables LLM reasoning in an unrestricted latent space instead of using natural language. By utilizing the last hidden state of the LLM as a representation of the reasoning state and feeding it back as the subsequent input embedding directly in the continuous space, Coconut demonstrates improved performance on reasoning tasks with fewer thinking tokens during inference. This approach leads to emergent advanced reasoning patterns, including the ability to encode multiple alternative next reasoning steps, allowing the model to perform a breadth-first search rather than committing to a single deterministic path.

Recent analysis [143] of implicit reasoning in transformers reveals important insights into its limitations. While language models can perform step-by-step reasoning and achieve high accuracy in both in-domain and out-of-domain tests via implicit reasoning when trained on fixed-pattern data, implicit reasoning abilities emerging from training on unfixed-pattern data tend to overfit specific patterns and fail to generalize further. These findings suggest that language models acquire implicit reasoning through shortcut learning, enabling strong performance on tasks with similar patterns while lacking broader generalization capabilities.

The evolution of unstructured reasoning approaches demonstrates the remarkable adaptability of language models to different reasoning paradigms. From simple prompting techniques to sophisticated implicit reasoning methods, these approaches leverage the inherent capabilities of LLMs to perform complex logical inferences without requiring explicit structural constraints. This flexibility enables more intuitive problem-solving while maintaining efficiency and effectiveness across diverse reasoning tasks.

2.2.3 Planning

Planning is a fundamental aspect of human cognition, enabling individuals to organize actions, anticipate outcomes, and achieve goals in complex, dynamic environments [144]. Formally, planning can be described as the process of constructing potential pathways from an initial state to a desired goal state, represented as $P : S_0 \rightarrow \{a_1, a_2, \dots, a_n\} \rightarrow S_g$, where S_0 is the starting state, $\{a_1, a_2, \dots, a_n\}$ denotes a sequence of possible actions, and S_g is the goal state. Unlike direct reasoning, planning involves generating hypothetical action sequences before execution, functioning as computational nodes that remain inactive until deployed. This cognitive ability emerges from the interplay of specialized neural circuits, including the prefrontal cortex, which governs executive control, and the hippocampus, which supports episodic foresight and spatial mapping. Insights from decision theory, psychology, and cybernetics—such as rational frameworks, prospect theory, and feedback loops—demonstrate how planning allows humans to transcend reactive behavior, actively shaping their futures through deliberate intent and adaptive strategies. This capacity not only underpins intelligent behavior but also serves as a model for developing LLM-based agents that seek to replicate and enhance these abilities computationally [145, 146].

In human cognition, planning operates as a hierarchical process, integrating immediate decisions with long-term objectives. This reflects the brain’s modular architecture, where neural systems collaborate to balance short-term demands with future possibilities—a dynamic informed by control theory’s principles of stability and optimization. Similarly, LLM-based agents employ planning by leveraging their extensive linguistic knowledge and contextual reasoning to transform inputs into actionable steps. Whether addressing structured tasks or unpredictable challenges,

these agents emulate human planning by decomposing objectives, evaluating potential outcomes, and refining their strategies—blending biological inspiration with artificial intelligence. This section examines the theoretical foundations and practical techniques of planning, from sequential approaches to parallel exploration, highlighting its critical role in intelligent systems.

Despite the potential of LLMs in automated planning, their performance faces limitations due to gaps in world knowledge [147]. LLMs often lack deep comprehension of world dynamics, relying on pattern recognition rather than genuine causal reasoning, which hinders their ability to manage sub-goal interactions and environmental changes [148]. Additionally, their reliance on static pre-training data restricts adaptability in real-time scenarios, limiting their generalization in dynamic planning tasks [149]. The absence of an intrinsic System 2 reasoning mechanism further complicates their ability to independently generate structured, optimal plans [150]. However, researchers have proposed strategies such as task decomposition, search optimization, and external knowledge integration to mitigate these challenges.

Task Decomposition Task decomposition enhances LLM planning by breaking complex goals into smaller, manageable subtasks, reducing problem complexity and improving systematic reasoning. The Least-to-Most Prompting method [138] exemplifies this approach, guiding LLMs to solve subproblems incrementally. ADaPT [151] further refines this strategy by dynamically adjusting task decomposition based on complexity and model capability, particularly in interactive decision-making scenarios. These methods also facilitate parallel subtask processing, backward error tracing, and independence determination [132], providing a structured framework for reasoning.

In LLM planning, tasks function as executable units—distinct from static state descriptions in formal models—emphasizing structured sequences that achieve intended outcomes [66]. These tasks vary in nature: some are subproblems requiring specific solutions (e.g., solving equations within broader challenges), while others involve tool invocation (e.g., querying APIs for weather data in travel planning) [152, 153]. Alternatively, tasks may be represented as graph nodes mapping dependencies, such as prioritizing objectives in project management [154]. By defining clear, modular goals, these formulations enhance reasoning and action efficiency, guiding agents through complex problem spaces with greater precision [93].

Searching Given the stochastic nature of LLMs [155], parallel sampling combined with aggregated reasoning can improve inference performance. Task decomposition structures individual solution trajectories, enabling the construction of a solution space that includes multiple pathways to a goal and their interrelationships [72, 156]. This space allows sampling diverse potential solutions [157], facilitating exploration through techniques like reflection, review, and parallel sampling informed by existing knowledge [158].

Computational constraints often preclude exhaustive evaluation, making efficient navigation of the solution space essential. Methods include tree search algorithms like LATS [159], heuristic approaches such as PlanCritic’s genetic algorithms [160], and CoT-SC, which identifies recurring solutions via self-consistency checks [78]. Reward-based models like ARMAP assess intermediate and final outcomes to optimize planning [106]. This iterative exploration and refinement process enhances adaptability, ensuring robust strategies for complex problems.

World Knowledge Effective planning requires agents to navigate dynamic environments, anticipate changes, and predict outcomes, underscoring the importance of world knowledge. RAP [74] examines the interplay between LLMs, agent systems, and world models, positioning LLMs as dual-purpose entities: as world models, they predict state changes following actions [107, 161]; as agents, they select actions based on states and goals [70]. This framework mirrors human cognition—simulating action consequences before selecting optimal paths—and unifies language models, agent models, and world models as pillars of machine reasoning [162].

Agents augment LLM capabilities by integrating external knowledge, addressing gaps in world understanding. ReAct employs an action-observation loop to gather environmental feedback, combining real-time data with linguistic knowledge to improve decision-making in complex scenarios [70]. This enables LLMs to iteratively refine their world models during action execution, supporting adaptive planning. Conversely, LLM+P [163] integrates LLMs with the PDDL planning language, converting natural language inputs into formalized representations solved by classical planners [164, 165]. This hybrid approach compensates for LLMs’ limitations in structured planning, merging their linguistic flexibility with the reliability of traditional systems.

Further advancements enhance LLM planning through world knowledge integration. CodePlan [166] uses code-form plans—pseudocode outlining logical steps—to guide LLMs through complex tasks, achieving notable performance improvements across benchmarks [167]. The World Knowledge Model (WKM) equips LLMs with prior task knowledge and dynamic state awareness, reducing trial-and-error and hallucinations in simulated environments [168]. A neuro-symbolic approach combining Linear Temporal Logic with Natural Language (LTL-NL) integrates formal logic with

LLMs, leveraging implicit world knowledge to ensure reliable, adaptive planning [169]. Together, these methods illustrate how structured frameworks and environmental understanding can transform LLMs into effective planners.